

---

# ANALYSIS ON AN IMPLEMENTATION OF THE GENS-DOMINGOS SUM-PRODUCT NETWORK STRUCTURAL LEARNING SCHEMA

---

**Renato Lui Geh**

Computer Science

Institute of Mathematics and Statistics

University of São Paulo

renatolg@ime.usp.br

**ABSTRACT.** Sum-Product Networks (SPNs) are a class of deep probabilistic graphical models. Inference in them is linear in the number of edges of the graph. Furthermore, exact inference is achieved, in a valid SPN, by running through its edges twice at most, making exact inference linear. The Gens-Domingos SPN Schema is an algorithm for structural learning on such models. In this paper we present an implementation of such schema, analyzing its complexity, discoursing implementational and theoretical details, and finally presenting results and experiments achieved with this implementation.

**Keywords** cluster analysis; data mining; probabilistic graphical models; tractable models; machine learning; deep learning

## 1. INTRODUCTION

A Sum-Product Network (SPN) is a probabilistic graphical model that represents a tractable distribution of probability. If an SPN is valid, then we can perform exact inference in time linear to the graph's edges. Its syntax is different to other conventional models (read bayesian and markov networks) in the sense that its graph does not explicitly model events and (in) dependencies between variables. That is, whilst variables in a bayesian network are represented as nodes in the graph, with each edge connecting two nodes asserting a dependency relationship between the connected variables, a node in an SPN may not necessarily represent a variable or event, neither an edge connecting two nodes represent dependence. In this sense, SPNs can be seen as a type of probabilistic Artificial Neural Network (ANN). However, whilst neural networks represent a function, SPNs model a tractable probability distribution. Furthermore, SPNs are distinct from standard neural networks seeing that, whereas ANNs have only one type of neuron with an activation function mapping to values in  $[0, 1]$ , SPNs have two kind of neurons, which we will see in the next sections. Still, SPNs retain certain important characteristics from ANNs as we will discuss later, with mainly its deep structure properties as the most interesting feature.

The Gens-Domingos Schema [GD13], or **LearnGD** as we will reference it throughout this paper, is an SPN structural learning algorithm proposed by Robert Gens and Pedro Domingos. Gens and Domingos call it a schema because it only provides a template of what the algorithm should be like. We will discuss **LearnGD** in details in the next section. This paper documents a particular implementation of the GD schema. Other implementations may have different results.

In this document, we show how we implemented the **LearnGD** algorithm. We analyse the complexity of each algorithm component in detail, later referring to such analyses when drawing conclusions on the overall complexity of the algorithm. As we have mentioned before, since the **LearnGD** schema depends heavily on implementation, the complexity we achieve in this particular case may differ from other implementations. After each analysis, we then look at the algorithm as whole, drawing conclusions on time and memory usage, as well as implementation details that could potentially decrease the algorithm runtime. We also comment on how to implement better concurrency then how it is currently coded in our implementation. We then show some results on experiments made on image classification and image completion.

## 2. SUM-PRODUCT NETWORKS

In this section we will define SPNs differently from other articles [GD13; PD11; DV12] as the original more convoluted definition is of little use for the **LearnGD** algorithm. Our definition is almost identical to the original **LearnGD** article [GD13], with the exception that we assume that an SPN is already normalized. This fact changes nothing, since Peharz *et al* recently proved that normalized SPNs have as much representability power as unnormalized SPNs [Peh+15]. Before we enunciate the formal definition of an SPN, we will give an informal, vague definition of an SPN in order to explain what completeness, consistency, validity and decomposability — which are an important set of definitions — of an SPN mean.

A sum-product network represents a tractable probability distribution through a DAG. Such digraph must always be weakly connected. A node can either be a leaf, a sum, or a product node. The scope of a node is the set of all variables present in all its descendants. Leaf nodes are tractable probability distributions and their scope is the scope of its distribution, sum nodes represent the summing out of the variables in its scope and product nodes act as feature hierarchy. We refer to a sub-SPN  $S$  rooted at node  $i$  as  $S(i)$ , while the SPN rooted at its root is denoted as  $S(\cdot)$  or simply  $S$ . The scope of a node will be denoted as  $Sc(i)$ , where  $i$  is a node. The set of children of a node will be denoted as  $Ch(i)$ . Similarly,  $Pa(i)$  is the set of parents of node  $i$ .

**Definition 2.1** (Completeness).

Let  $S$  be an SPN and  $r$  be its root.  $S$  is complete iff  $Sc(i) = Sc(j), i \neq j; \forall i, j \in Ch(r)$ .

**Definition 2.2** (Consistency).

Let  $S$  be an SPN,  $r$  be its root and  $X$  a variable in  $Sc(r)$ .  $S$  is consistent iff  $X$  takes the same value in all product nodes in  $S$ .

**Definition 2.3** (Validity).

*An SPN  $S$  is valid iff it always computes the correct probability of evidence  $S$  represents.*

**Theorem 2.1.** *An SPN  $S$  is valid if it is both complete and consistent.*

## REFERENCES

- [DV12] Aaron Dennis and Dan Ventura. “Learning the Architecture of Sum-Product Networks Using Clustering on Variables”. In: *Advances in Neural Information Processing Systems* 25 (2012).
- [GD13] Robert Gens and Pedro Domingos. “Learning the Structure of Sum-Product Networks”. In: *International Conference on Machine Learning* 30 (2013).
- [PD11] Hoifung Poon and Pedro Domingos. “Sum-Product Networks: A New Deep Architecture”. In: *Uncertainty in Artificial Intelligence* 27 (2011).
- [Peh+15] Robert Peharz et al. “On Theoretical Properties of Sum-Product Networks”. In: *International Conference on Artificial Intelligence and Statistics 18 (AISTATS 2015)* (2015).